TUBA Working Group K. Robert Glenn (NIST)
INTERNET-DRAFT

Integrated Network Layer Security Protocol For TUBA

(draft-ietf-tuba-inlsp-00.txt)

(Posted: May 16, 1994/Expires: November 16, 1994)

Status of This Memo

Abstract

This Internet Draft specifies a protocol that may be used by End Systems (ESs) and Intermediate Systems (ISs) in order to provide security services in support of TUBA. The TUBA deployment and transition plan relies on a dual-stack (i.e., CLNP and IPv4) approach to evolving the Internet to IPng. Thus, to provide secure operation in an TUBA environment this Internet Draft defines a simple Integrated Network Layer Security Protocol (I-NLSP) that provides confidentiality and integrity services for both CLNP and IPv4. TUBA systems supporting I-NLSP will be capable of secure operations with: (1) other TUBA/I-NLSP systems using either CLNP or IP, and or (2) existing IPv4 operating behind TUBA/I-NLSP capable secure ISs.

Contents

1. Introduction

The capability for "secure operation" is identified [IPng-Criteria] as a required integral component of any IPng proposal. It is also widely recognized that the evolution to any IPng may be slow and will require IPng to coexist and inter-operate with IPv4 systems for an extended period of time. As such, the security mechanisms of an IPng must address their coexistence and inter-operation with IPv4 systems also.

This Internet Draft specifies a protocol that may be used by End Systems (ESs) and Intermediate Systems (ISs) in order to provide security services in support of TUBA. The TUBA deployment and transition plan relies on a dual-stack (i.e., CLNP and IPv4) approach to evolving the Internet to IPng. Thus, to provide secure operation in an TUBA environment this Internet Draft defines a simple Integrated Network Layer Security Protocol (I-NLSP) that provides confidentiality and integrity services for both CLNP and IPv4. TUBA systems supporting I-NLSP will be capable of secure operations with: (1) other TUBA/I-NLSP systems using either CLNP or IP, and or (2) existing IPv4 operating behind TUBA/I-NLSP capable secure ISs.

It should be noted that I-NLSP may be suitable for other, non-TUBA related, scenarios (e.g., implementation and general use in "IPv4 only" and "CLNP only" systems, extensions to support other connectionless network protocols). These other applications of I-NLSP are beyond the scope of this document.

This Internet Draft specifies the following services:

1. Connectionless Confidentiality: Access to information is restricted to authorized individuals, entities, and processes. Confidentiality is provided by encrypting the information requiring protection.

2. Connectionless Integrity: Information cannot be altered without detection. Integrity is provided by calculating a secured checksum over the information requiring protection.

Although the degree of protection afforded by some security services depends on the use of some specific cryptographic techniques, correct operation of this protocol is not dependent on the choice of a particular integrity or confidentiality mechanisms; that is left as a local matter for the communicating systems.

Only those ESs and ISs requiring secure communications will be required to alter their existing IP or CLNP implementations. ESs

running IPv4 or CLNP without I-NLSP will be able to continue communicating with other ESs and ISs running IPv4 or CLNP with or without I-NLSP in a non-protected fashion. ISs with existing implementations of IPv4 or CLNP will not require any changes to be able to forward datagrams protected by I-NLSP.

The remainder of this Internet Draft contains a functional description of the I-NLSP protocol and it's components. Also included, are Appendices that contain descriptions of local security policy mechanisms; descriptions and contents of globally known tables to be used by I-NLSP; and an example Key/Security Association Exchange protocol to be used with I-NLSP.

2. Functional Overview of I-NLSP

I-NLSP supports the ability to transfer protected or unprotected connectionless datagrams between peer DFP ESs and ISs. Protection is defined as the application of one or more of the security services offered by I-NLSP. I-NLSP resides in the Network Layer and is a functional component of the Datagram Forwarding Protocol (DFP) as shown in Figure 1. The term DFP is used throughout this document to generically represent the services offered by either IP or CLNP.

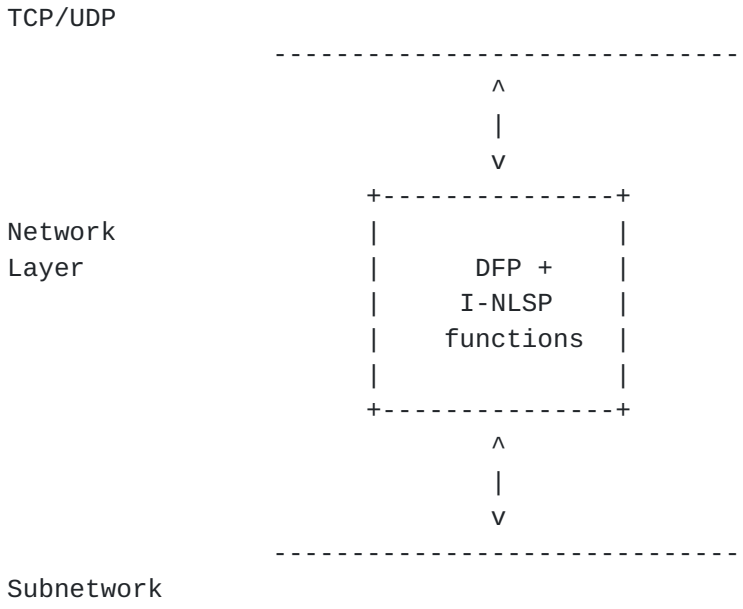

Figure 1: I-NLSP within the Network Layer

I-NLSP performs two functions, Encapsulation and Decapsulation. Within the Network Layer, there are two distinct modes of operation

for which these functions are performed, ES Mode and IS Mode. Within I-NLSP there are two modes of Encapsulation/Decapsulation, TCP/UDP Encapsulation/Decapsulation Mode and DFP Encapsulation/Decapsulation Mode. The following sections provide an overview of these four modes of operation.

2.1. ES Mode

When the DFP receives data from TCP/UDP to be forwarded, local security policy is checked to determine if I-NLSP security services are required for the destination address. Local security policy dictates whether non-protected communication with the destination is permitted. If these services are required, I-NLSP functions are invoked. I-NLSP generates a Secure Data Transfer (SDT) PDU and encapsulates the data within the SDT PDU. The Encapsulation Function applies either integrity, confidentiality, or both depending on local security policy. The SDT PDU becomes the payload of a DFP datagram and is forwarded towards the peer I-NLSP Entity (I-NLSPE).

When the DFP receives data from the subnet, local security policy and the DFP header are checked to determine if I-NLSP security services have been applied. Local security policy dictates whether non-protected communication with the source is permitted. If I-NLSP security services have been applied, I-NLSP functions are invoked. I-NLSP decapsulates the SDT PDU. The Decapsulation Function checks for integrity, confidentiality, or both depending on local security policy. The decapsulated data is then passed to TCP/UDP or treated as a new DFP packet depending on which mode of Encapsulation was used.

2.2. IS Mode

When the DFP receives a DFP packet from the subnet to be forwarded, local security policy is checked to determine if I-NLSP security services are required for the destination address. Local security policy dictates whether non-protected communication with the destination is permitted. If these services are required, I-NLSP functions are invoked. I-NLSP generates a SDT PDU and encapsulates the data within the SDT PDU. The Encapsulation Function applies either integrity, confidentiality, or both depending on local security policy. The SDT PDU becomes the payload of a DFP datagram and is forwarded towards the peer I-NLSPE.

When the DFP receives data from the subnet, local security policy and

the DFP header are checked to determine if I-NLSP security services have been applied. Local security policy dictates whether non-protected communication with the source is permitted. If I-NLSP security services have been applied, I-NLSP functions are invoked. I-NLSP decapsulates the SDT PDU. The Decapsulation Function checks for integrity, confidentiality, or both depending on local security policy. The decapsulated data is then forwarded toward its final destination.

2.3. TCP/UDP Encapsulation/Decapsulation Mode

TCP/UDP Encapsulation/Decapsulation mode dictates that the SDT PDU contains TCP/UDP Data. The remainder of this section explains the cases in which this mode is used.

When an ES I-NLSPE is communicating with another ES I-NLSPE, it is preferable for the I-NLSPE to only encapsulate the TCP/UDP data and avoid the overhead generated by the DFP Encapsulation/Decapsulation Mode. IS I-NLSPEs communicating with ES I-NLSPEs could also use this mode but there are problems associated with fragmentation before encapsulation. As a result of these problems it is recommended that ISs always use the DFP Encapsulation/Decapsulation Mode.

2.4. DFP Encapsulation/Decapsulation Mode

DFP Encapsulation/Decapsulation mode dictates that the SDT PDU contains an entire DFP packet. The remainder of this section explains the cases in which this mode is used.

When an I-NLSPE (ES or IS) is communicating with an IS I-NLSPE two destination addresses are required (one to get the DFP packet to the IS and another to get the packet to the destination ES). By encapsulating an entire DFP packet, both destination addresses are preserved. Encapsulating the entire DFP packet also preserves fragmentation information kept in the DFP packet header, allowing the encapsulated data to be reassembled by the peer I-NLSPE.

2.5. Security Association

A Security Association(SA) is defined as a relationship between communicating lower layer entities for which there exists a common set of corresponding attributes. These SA attributes define the

particular mechanisms and how they are to be used by I-NLSP to provide security services between peer I-NLSPEs. In order to protect an instance of connectionless communication, an existing suitable SA is used. If no suitable SA exists, one needs to be established between the communicating parties. The Security Association may be established "out-of-band" or using an "in-band" SA Protocol. A complete SA Protocol is outside the scope of this Internet Draft. A partially defined example is provided in Appendix C.

3. Security Association Attributes

The following SA Attributes control the operation of I-NLSP and the mechanisms used to provide protection. The associated values to these attributes shall be set up on SA establishment. The SA Attributes are to be stored in some form of database or table, uniquely indexed by DFP destination addresses Security Association Identifiers (SAIDs). The SA Attributes are assumed to be symmetric between the peer I-NLSPEs.

Some attributes have recommended default values. I-NLSP is generic by nature and leaves a large number of decisions up to local security policy and implementation. The default values are provided to: suggest a minimal set of security services which I-NLSP should provide; aid implementors in providing a core set of functionality in their products that will enable interoperability; and initiate the process of define a globally known table of security algorithms along with their associated attributes.

1. SA Identification:

   - SAID: Integer of range 0..65535

     Coupled with DFP Destination address, uniquely identifies the current SA established with peer I-NLSP entity.

     Note: Some values have been reserved to identify globally unique SAID values (see Appendix B).

2. DFP Address of peer I-NLSP entity:

   - Peer_Adr: Address of format specified by DFP.

3. DFP Address of entities served through the remote peer:

   - Adr_Served: Set of Addresses of format specified by DFP. (Default: Peer_Adr)

4. Flags:

- o Confidentiality: Boolean (Default: false)

  Flag used to determine if confidentiality is required.

- o Integrity: Boolean (Default: true)

  Flag used to determine if integrity is required.

5. ICV mechanism attributes:

- o ICV_Alg: Object Identifier

  The value of this attribute shall be an index into an globally known table of security algorithms. This attribute implies certain attributes of the integrity mechanism such as separate generate and check algorithms, initialization vectors, block size, etc.

  (Default: Index for DES CBC)

- o ICV_Length: Integer of range 1..8.

  Specifies the length of the ICV.

  (Default: 8 octets)

- o ICV_Gen_Key: length and form defined by ICV_Alg.

- o ICV_Check_Key: length and form defined by ICV_Alg.

  (Default: ICV_Gen_Key)

6. Confidentiality Mechanism Attributes:

- o Enc_Alg: Object Identifier

  The value of this attribute shall be an index into an globally known table of security algorithms. This attribute implies certain attributes of the confidentiality mechanism such as separate encryption and decryption algorithms, initialization vectors, block size, etc.

  (Default: Index for DES CBC)

- o Data_Enc_Key: length and form defined by Enc_Alg.

o Data_Dec_Key: length and form defined by Enc_Alg.

(Default: Data_Enc_Key)

4. Secure Data Transfer PDU Format

All SDT PDUs shall contain an integral number of octets. The format of a Secure Data Transfer PDU shall be as shown in Figure 2. Security services are applied to the SDT PDU such that the integrity mechanism is applied to the SDT PDU Header and portions of the Protected-Octet-String. The confidentiality mechanism is applied to portions of the Protected-Octet-String.

Figure 2: Secure Data Transfer PDU Structure

4.1. SDT PDU Header

The format of the Header shall be as shown in Figure 3. Bit positions are denoted as integers above the diagram.

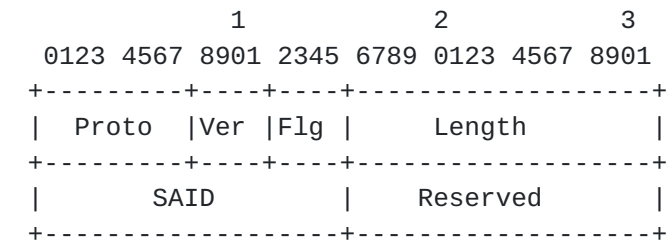

Figure 3: SDT PDU Header

1. Proto - This field contains the protocol of the DFP user (TCP/UDP). The length of this field is 8 bits.

2. Ver - This field contains the version number of the protocol represented by this SDT PDU. The length of this field is 4 bits.

3. Flg - This field contains optional flags used by I-NLSP

in processing the SDT PDU. The length of this field is 4 bits No values have been idenfied at this time. This field is set to zero when sending and ignored upon receipt.

4. Length - This field contains the length, in octets, of the Protected-Octet-String before the application of the confidentiality mechanism. The length of this field is 16 bits. It is possible for an encryption algorithm to append extra octets to the Protected-Octet-String for its own purposes. The Length field is not modified to reflect this.

5. SAID - The SAID field shall contain the Security Association Identifier used to identify this particular SA. The length of this field is 16 bits.

6. Reserved - This field is reserved for future use. This field is set to zero when sending and ignored upon receipt.

4.2. Protected_Octet_String

The Protected_Octet_String field contains the data which has been protected by the I-NLSP security mechanisms. The format of this field, is dependent on which security mechanisms are to be applied. Figure 4 shows the format of the Protected_Octet_String when only Integrity is applied. In this case it is imperative that the ICV mechanism protects the ICV from alteration.

When confidentiality is applied, most of the resulting Protected_Octet_String is perceived as a random octet string with no distinguishable characteristics. Figure 5 shows the format of the Protected_Octet_String before confidentiality is applied and integrity is not to be applied. Figure 6 and 7 shows the format of the Protected_Octet_String the application of integrity and confidentiality.

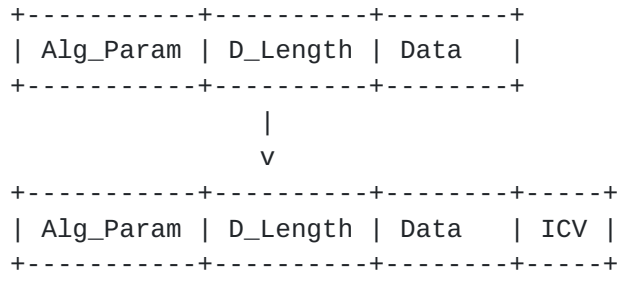

Figure 4. Protected_Octet_String Before and After Integrity.

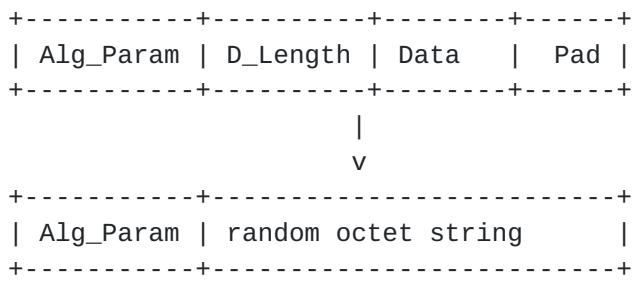

Figure 5. Protected_Octet_String Before and After Confidentiality.

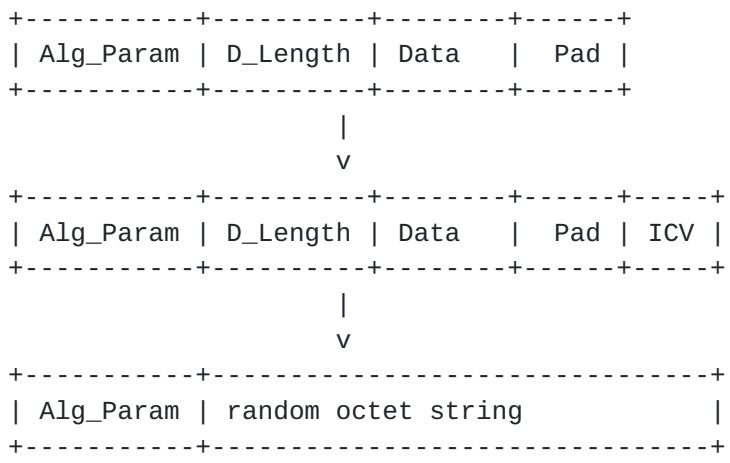

Figure 6. Protected_Octet_String Before and After Integrity, Before and After Confidentiality.

1. Alg_Param - This field contains information required by the specific integrity and confidentiality algorithms used in protecting the SDT PDU Data (eg., initialization vector). The length and format of this field are defined by the ICV_Alg and Enc_Alg.

2. D_Length - This field contains the length of the SDT PDU Data. The length of this field is two octets.

3. Data - This field contains the data that is to be encapsulated.

4. Pad - This field contains a string of octets with locally defined values. This field is used by the confidentiality mechanism to pad to required lengths.

5. ICV - This field contains the Integrity Check Value (ICV). The length of this field shall be defined by the ICV Length contained in the Security Association attributes.

5. I-NLSP Functional Description

The following sections describe the functionality of I-NLSP. It is assumed that local security policy and implementation will determine how and when I-NLSP encapsulation is to be applied. Appendix A suggests mechanisms to aid the DFP with this decision.

For encapsulation, the I-NLSPE determines the security policy and services to be applied to the data; encapsulates the data in the form of a SDT PDU; and forwards to the peer I-NLSPE. For decapsulation, the I-NLSPE determines the security policy and services that were applied to the data; decapsulates the data; and sinks or forwards the data depending on the appropriate mode (ES or IS). Figures 7 and 8 are included to show functional flow and SDT PDU encapsulation. Figures 9 and 10 are included to show functional flow and SDT PDU decapsulation.

At many points in the following sections, the I-NLSPE checks that some condition is satisfied. Unless otherwise specified, whenever such a check fails, the I-NLSPE shall discard the DFP data currently being processed. Optionally, the entity may also file an audit/error report. Which failures to be audited is considered to be a local matter. Throughout the following sections, this procedure is known as the error process.

5.1. Encapsulation Function

Figure 7 and the associated sections describe in detail the steps required to encapsulate data in a protective envelope.

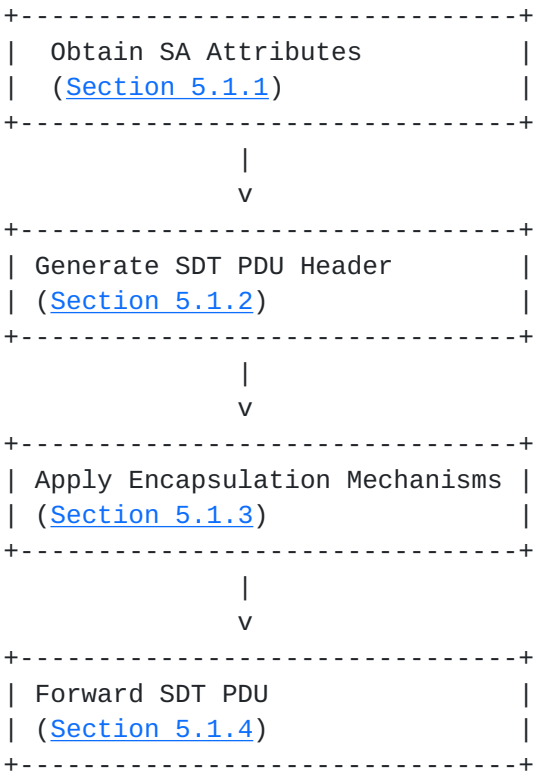

Figure 7. Functional Flow of Encapsulation Request

5.1.1. Obtain SA Attributes:

1. I-NLSP shall check that a Security Association has been established between this I-NLSPE and the peer I-NLSPE. The DFP Destination Address is compared against the Adr_Served SA Attribute list. The DFP Destination of the Peer I-NLSPE is then found in the associated Peer_Adr SA attribute.

2. If an association is found and the association does not specify either Integrity or Confidentiality, the data is forwarded normally. Whether or not this is permitted is determined by local security policy.

3. If an association is found and the association does specify either Integrity, Confidentiality, or both, proceed to the next step.

4. If no association is found but an "in-band" SA establishment is supported through a Security Association Protocol (SA-P), attempt to establish a security association with the Destination within a given, locally defined, timeout period is made. If the "in-band" SA establishment is successful, the data is processed as if an association has been discovered.

5. If no association is found and a SA cannot be established "in band", I-NLSP will perform the error process described above.

5.1.2. Generate SDT PDU Header

1. Determine Encapsulation Mode (TCP/UDP or DFP).

   - If the DFP Destination Address is equal to Peer_Adr and the data to be encapsulated is not a fragmented DFP packet, then use TCP/UDP Encapsulation Mode.

   - If the DFP Destination Address is not equal to Peer_Adr or the data to be encapsulated is a fragmented DFP packet, then use DFP Encapsulation Mode.

2. The Proto field is set to the value appropriate to the Encapsulation Mode.

   - TCP/UDP for TCP/UDP Encapsulation Mode.

   - DFP for DFP Encapsulation Mode.

3. The Ver field is set to 0x01.

4. The Flg and reserved fields are set to zero.

5. The SAID is set to the SAID SA Attribute identifying the SA found in section 5.1.1.

6. The Length field is set to D_Length + ICV_Length + 2 (length of D_Length).

5.1.3. Apply Encapsulation Mechanisms

Applying encapsulation mechanisms involves ICV generation and data encryption. The Integrity flag SA attribute is used to determine whether an ICV is generated and included in the SDT PDU. The Confidentiality flag SA attribute is used to determine whether confidentiality is applied to the SDT PDU. The following steps are taken, in order, to generate the complete encapsulated SDT PDU. Errors are processed as described above.

1. If Confidentiality is true, Alg_Param and Pad are added to the SDT PDU as needed. The length of these fields are added to the Length field of the SDT PDU Header.

2. If Integrity is true, Alg_Param are added to the SDT PDU as needed. An ICV of length, ICV_Length is generated over the, Alg_Param, D_Length, Data, and Pad, using the ICV algorithm specified by ICV_Alg along with the ICV_Check_Key, If additional padding is required for ICV generation, a pad of zeroes is used. The ICV is appended to the SDT PDU.

3. If Confidentiality is true, the D_Length, Data, Pad, and ICV are encrypted using the confidentiality algorithm specified by Enc_Alg, along with the Data_Enc_Key.

5.1.4. Forward SDT PDU

The SDT PDU becomes the payload of a DFP datagram and is forwarded toward the peer I-NLSPE. The DFP Source Address is set to this I-NLSPE DFP Address. The DFP Destination Address is set to the I-NLSPE Peer_Addr.

5.1.5. Complete Encapsulation Diagram

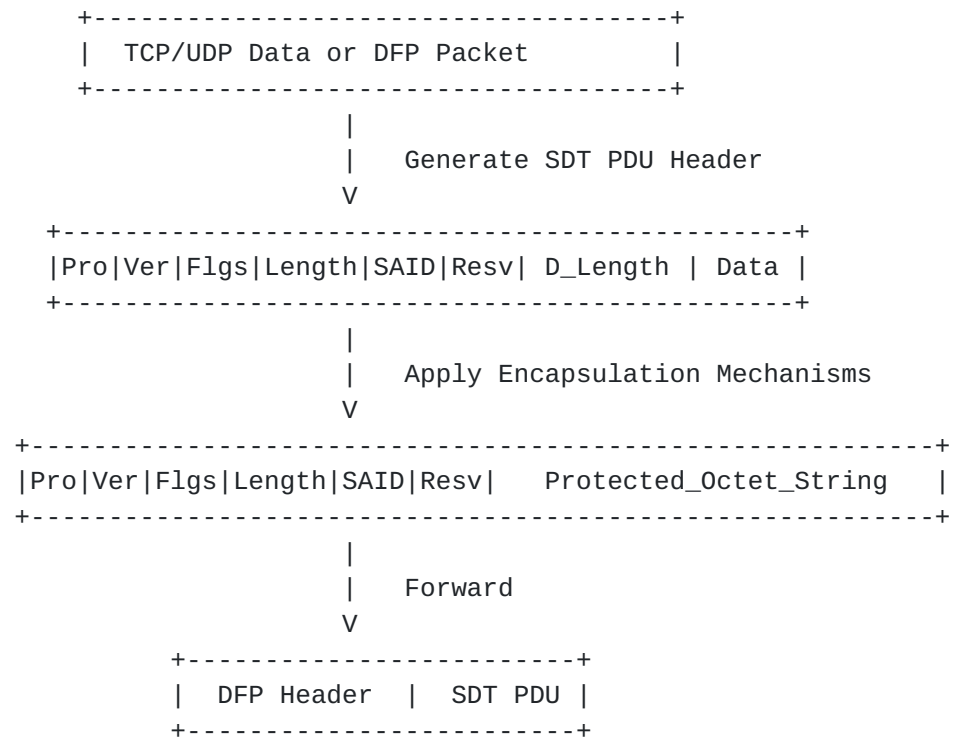

Figure 8: I-NLSP Encapsulation

5.2. Decapsulation Function

Figure 9 and the associated sections describe in detail the steps required to decapsulate data from the protective envelope. Errors and mechanism failures are processes as described above.

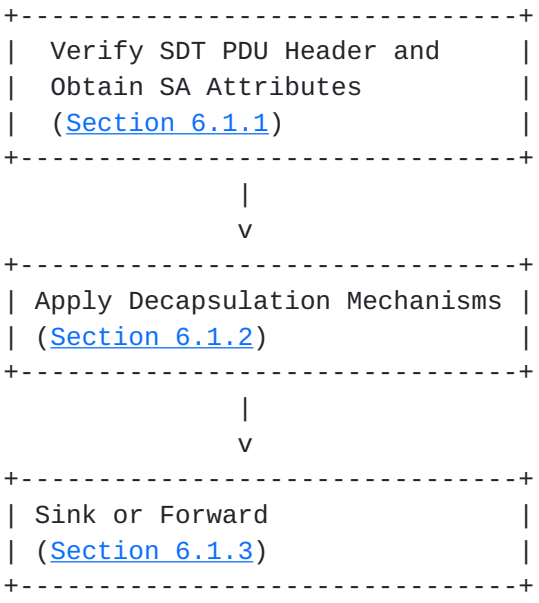

Figure 9. Functional Flow of Encapsulation Request

5.2.1. Verify SDT PDU Header and Obtain SA Attributes:

1. The Ver field must be set to 0x01.

2. The Proto field must be set to either TCP/UDP or DFP.

3. I-NLSP shall check that a Security Association has been established between this I-NLSPE and the Peer I-NLSPE. The DFP Source Address is compared with the Peer_Adr SA Attribute and the SAID from the SDT PDU Header is compared to the SAID associated with that Peer_Adr.

4. If an association is found, proceed to the next step.

5. If no association is found the I-NLSPE performs the error process described above.

5.2.2. Apply Decapsulation Mechanisms

Applying decapsulation mechanisms involves an ICV check and data decryption. The Integrity SA Attribute is used to determine whether an ICV was calculated and placed at the end of the SDT PDU. The Confidentiality SA attribute is used to determine whether confidentiality was applied to the SDT PDU. The following steps are taken to decapsulate the SDT PDU.

1. If Confidentiality is true,
   decipherment algorithm is applied to the
   Protected-Octet-String (excluding the Alg_Param field) using
   the decryption algorithm specified by the Enc_Alg, the
   Data_Dec_Key, and the parameters specified by the Alg_Param field.

2. If Integrity is true, the portion of the Alg_Param field
   provided by the Integrity mechanism is removed. The ICV
   check algorithm is performed on the SDT PDU Header, Alg_Param,
   D_Length, Data and Pad and Pad fields, using the ICV algorithm
   specified by ICV_Alg along with the ICV_Check_Key.
   If additional padding is required for the ICV
   check, a pad of zeroes is used.

5.2.3. Sink or Forward

If the SDT PDU Proto field was set to TCP/UDP the decapsulated data is sent to the TCP/UDP if operating in ES Mode or the data is forwarded if operating in IS Mode. If the SDT PDU Proto field was set to DFP the decapsulated data is processed as a newly received DFP datagram.

5.2.4. Decapsulation Diagram

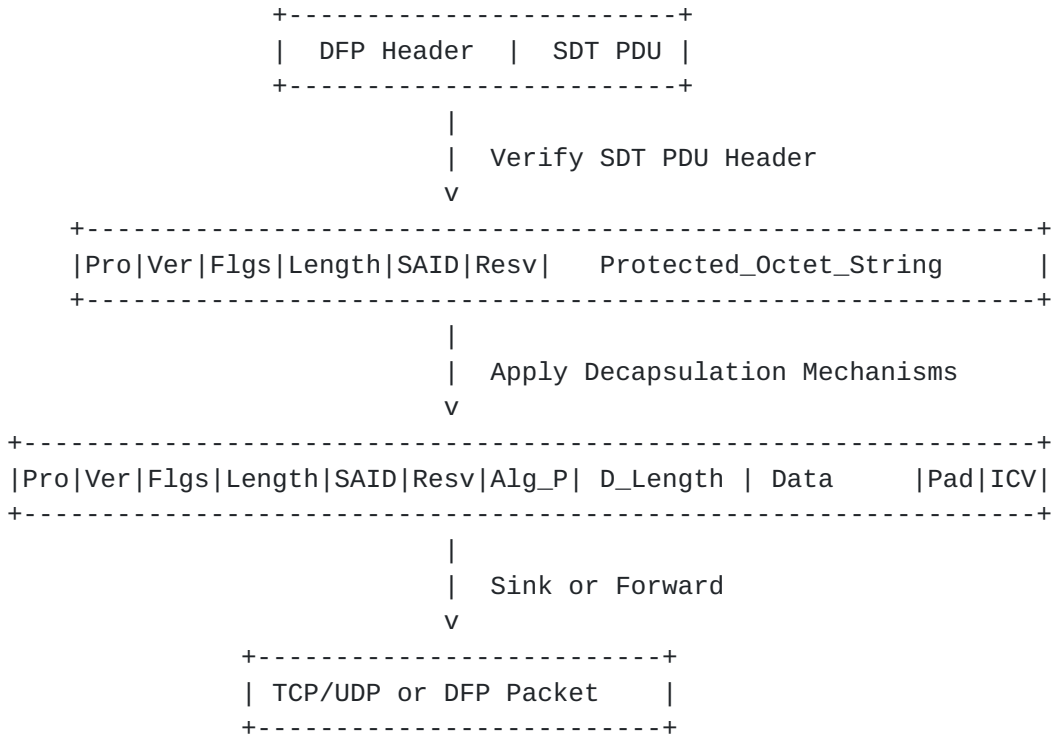

Figure 10: I-NLSP Decapsulation

6. IPv4 And I-NLSP

This section contains a functional description of how I-NLSP and IPv4 interoperate with each other. This functionality directly pertains to IPv4 and cannot be described in a more general way.

6.1. TCP/UDP Encapsulation/Decapsulation Mode

IPv4 uses the Protocol field that identifies the destination of the IPv4 data. At this time I-NLSP does not have an assigned protocol value. IN-PID is used to temporarily identify I-NLSP. I-NLSP uses the Proto field to identify the final destination of the SDT PDU data. Figure 11 provides an example of how this process works when the SDT PDU Data contains TCP data. In this example the Protocol field of the IPv4 header is set to the protocol value of I-NLSP (IN-PID) and the Proto field of the SDT PDU header is set to the protocol value of TCP (06) [RFC1340].

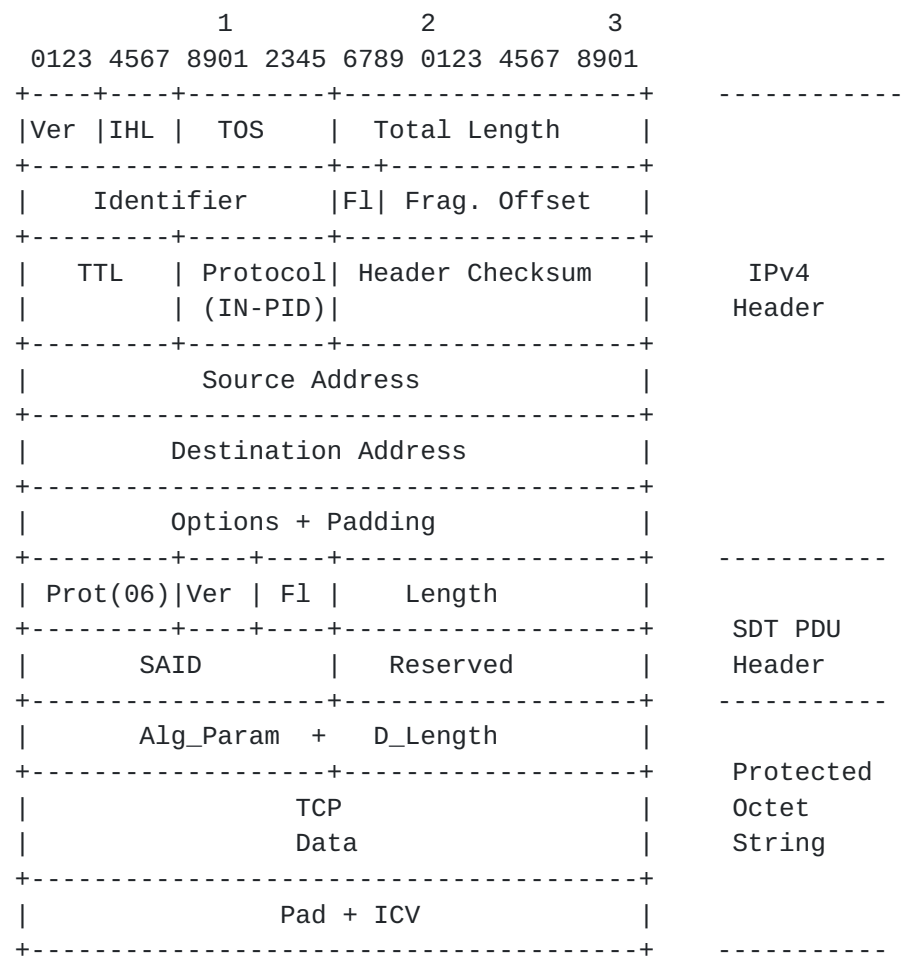

Figure 11. Encapsulated TCP Data as Payload of IPv4

6.2. DFP Encapsulation/Decapsulation Mode

Figure 12 provides an example of how this process works when the the SDT PDU Data contains an entire IPv4 packet. In this example the Protocol field of the outer most IPv4 header is set to the protocol value of I-NLSP (IN-PID); the Proto field of the SDT PDU header is set to protocol value representing IP-within-IP (94) [RFC1340]; and the Protocol field of the inner most IPv4 header is set to the protocol value representing TCP (06).

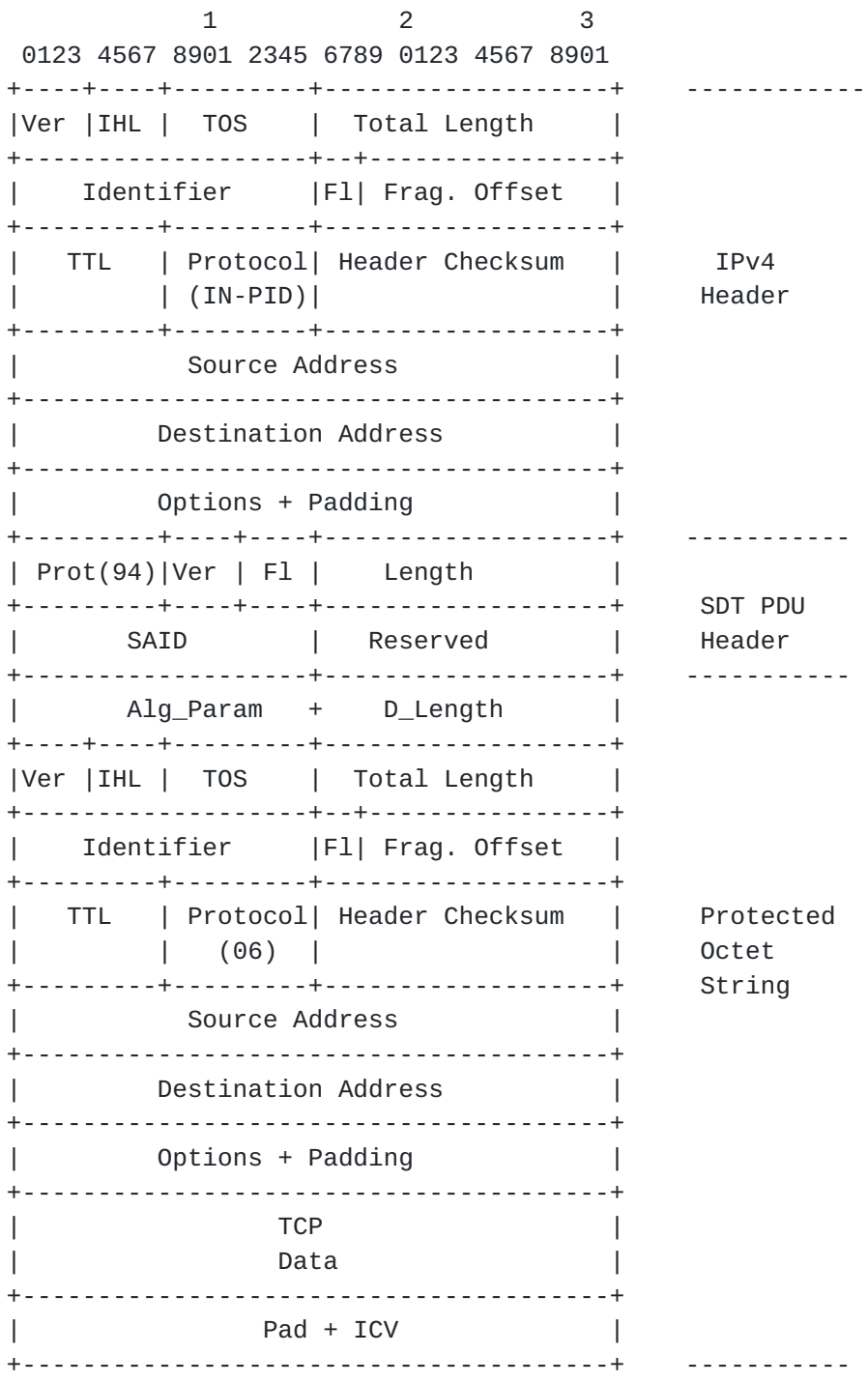

Figure 12. Encapsulated IPv4 Packet as Payload of IPv4

7. CLNP And I-NLSP

This section contains a functional description of how I-NLSP and CLNP interoperate. This functionality directly pertains to CLNP and cannot be described in a more general way.

7.1. TCP/UDP Encapsulation/Decapsulation Mode

CLNP uses the N-SEL portion of the Destination Address to identify the destination of the CLNP data. I-NLSP uses the Proto field to identify the final destination of the SDT PDU Data. Figure 13 provides an example of how this process works when the SDT PDU Data contains TCP data. In this example the N-SEL portion of the CLNP Destination Address is set to the protocol value of I-NLSP (IN-PID) and the Protocol field of the SDT PDU is set to the protocol value of TCP (06). Some of the following fields do not necessarily end on a 32bit boundary. A more complete description of the CLNP header can be found in [ISO8473], but is not required for this discussion.

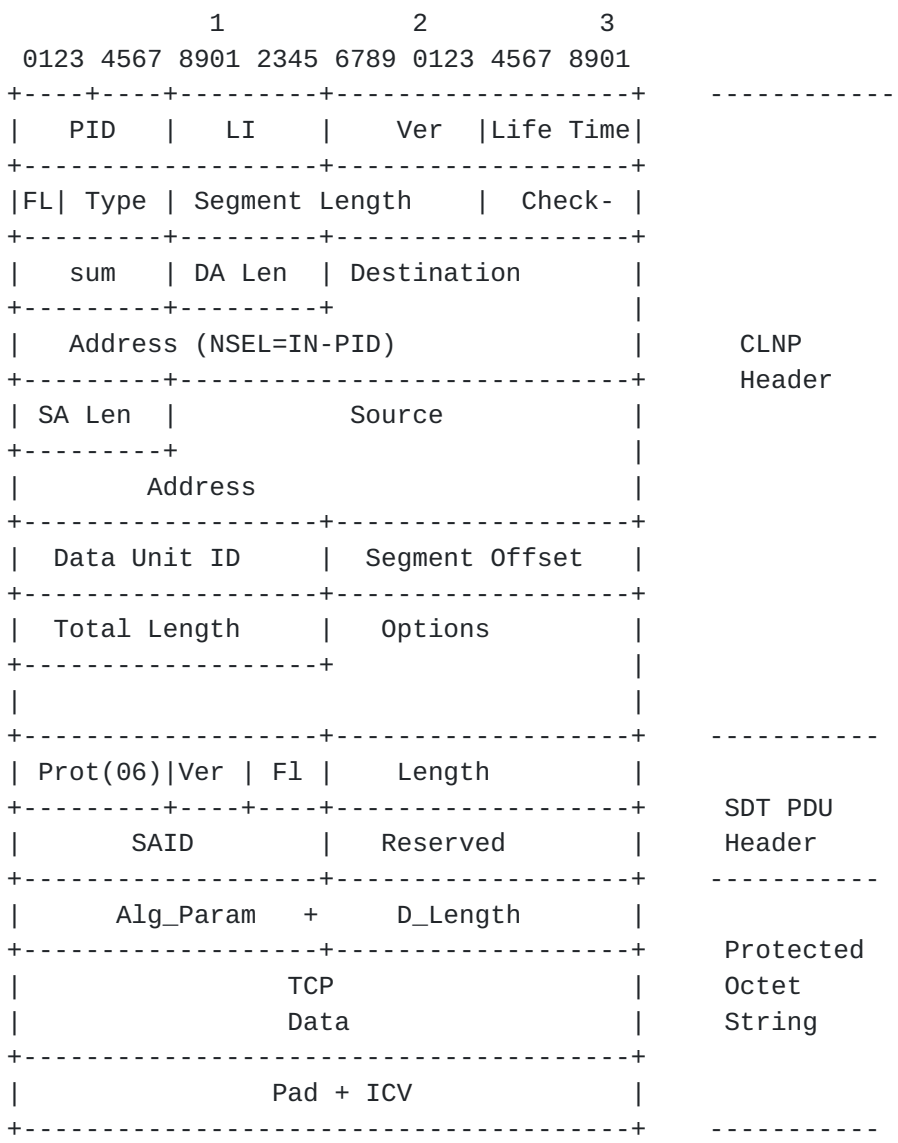

Figure 9. Encapsulated TCP Data as Payload of CLNP

7.2. DFP Encapsulation/Decapsulation Mode

Figure 14 provides an example of how this process works when the SDT PDU Data contains an entire CLNP PDU. In this example the N-SEL of the Destination Address of the outer most CLNP header is set to the protocol value of I-NLSP (IN-PID); the Prot field of the SDT PDU header is set to protocol value representing CLNP (129) [ISO8473]; and the N-SEL of the Destination Address of the inner most CLNP header is set to the protocol value representing TCP (06).

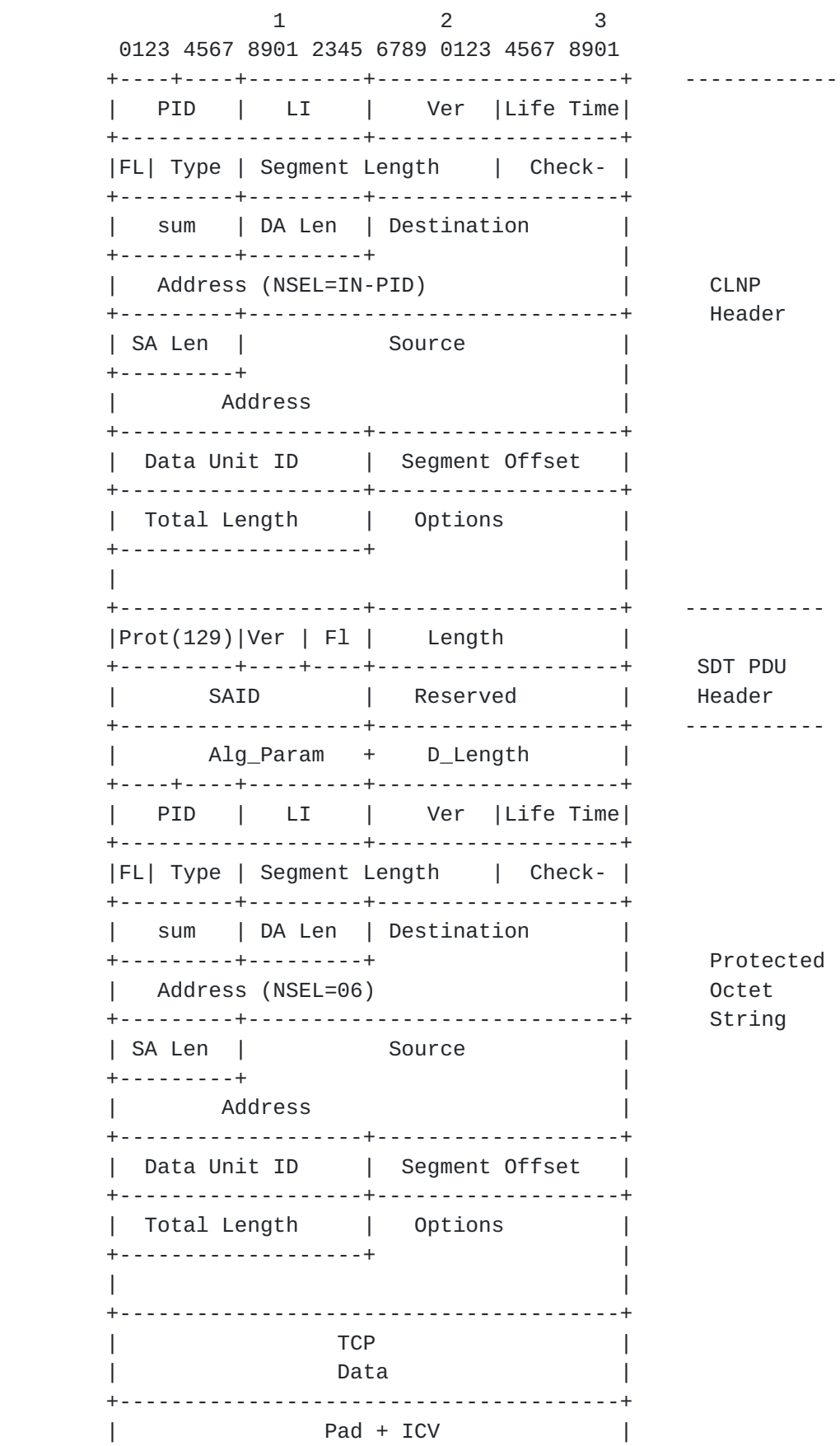
1 2 3
0123 4567 8901 2345 6789 0123 4567 8901
PID
LI
Ver
Life Time
FL
Type
Segment Length
Check-
sum
DA Len
Destination
Address (NSEL=IN-PID)
CLNP Header
SA Len
Source
Address
Data Unit ID
Segment Offset
Total Length
Options
Prot(129)
Ver
Fl
Length
SDT PDU Header
SAID
Reserved
Alg_Param
D_Length
PID
LI
Ver
Life Time
FL
Type
Segment Length
Check-
sum
DA Len
Destination
Address (NSEL=06)
Protected Octet String
SA Len
Source
Address
Data Unit ID
Segment Offset
Total Length
Options
TCP Data
Pad + ICV

Figure 10. Encapsulated CLNP Packet as Payload of CLNP

A. Policy Mechanisms

This section describes filters and flags used locally to help the DFP determine how and when I-NLSP security services are required.

- o Protected_Mode: Boolean (Default: false)

  Flag used to determine whether unprotected DFP packets are permitted.

- o Destination_Filter: Set of DFP Addresses

  Set of DFP Destination addresses which this I-NLSPE is not permitted to send unprotected DFP packets.

- o Source_Filter: Set of DFP Addresses

  Set of DFP Source addresses for which this I-NLSPE will provide I-NLSP services.

B. Tables

[Initial suggested values for reserved SAID values, Alg_IDs, etc.].

(TBD)

C. In-Band Security Association Exchange

[Initial description of a Security Association Exchange protocol using the Diffie-Helman algorithm. This section could potentially point to another Internet Draft on this subject].

(TBD)

References

[IPSEC] On-going Deliberations of the IETF IPSEC WG.

[IPng-Criteria] F. Kastenholz, C. Partridge, "Technical Criteria for Choosing IP:The Next Generation (IPng), Internet Draft, March 1994.

[ISO8473] ISO/IEC. Information Processing Systems - Data communications for providing the Connectionless-mode Network Service. ISO/IEC, 1988.

[ISO11577] ISO/IEC. Information technology - telecommunications and information exchange between systems - network layer security protocol. International Standard 11577, ISO/IEC JTC 1, USA, December 1993.

[RFC1340] J. Reynolds, J. Postel, "Assigned Numbers", Internet RFC 1340 June 1992.

[RFC1347] R. Callon, "TCP and UDP with Bigger Addresses (TUBA), A Proposal for Internet Addressing and Routing", Internet RFC 1347, June 1992.

[Stallings] William Stallings, Data and Computer Communication, Macmillan Publishing Company, Second Edition, 1988